Beyond Boolean: Uncharted Waters

Michael W. Toohey, Research Publications International,
and Brett Butler, INFOUR

ASIDIC Spring 1994 Meeting, Albuquerque, NM

A Patent Information Service Perspective on Advanced Searching

The Rapid Patent division of Research Publications International, an International Thomson company, is perhaps the most broadly based of the patent information service companies. RP provides U.S. and worldwide patents in formats ranging from simple copies to networked electronic service. Until recently, however, the firm's product offerings in the online sector were limited to contract search services using other vendors. Our work on behalf of our clients made us quite sensitive to the limits of traditional Boolean tools, which limits are all familiar to this audience.

Over the past year, Rapid Patent has been investigating (with the assistance of Brett Butler of Infour, an industry pioneer and entrepreneur) opportunities to provide more valuable search services to the unique patent research community. Not a traditional library market, patent searchers consist of patent lawyers and their firms; contract searchers which do onsite searches; engineers and inventors within and beyond industrial firms; and a limited number of general online search services firms or agencies.

We found that the U.S. electronic patent information industry was very much smaller than would be expected from the use of other business, scientific, and technical databases. It appears that this use is even more skewed to the large corporate entity than other online use, suggesting that the very small businesses which are the main fount of invention are increasingly denied good patent data access.

Our conclusion was that there was a very large untapped patent information market which could be developed ***if an easier search service product could be created for new users.*** We also observed, however, that the patent data itself is particularly difficult to conquer with key-word or Boolean search algorithms: it is perhaps the only literature written primarily to obfuscate rather than to inform.

At the time were completing this research, we became aware of a new system, MAYA, which had been developed in the research laboratory of Electronic Data Systems (EDS), the systems company which is part of General Motors.

The MAYA Search Engine

MAYA has been developed by EDS over a period of ten years. At the time, no parallel supercomputer systems were in commercial production - so EDS designed an architecture, built an operating system, and wrote the information retrieval application.

After its initial development, the R&D lab wanted to use a publicly available fulltext database to test its design in a very large file environment. EDS licensed the U.S. Patent Office fulltext database as a testbed for its system, and has spent considerable energy in testing its effectiveness using patent data.

EDS maintains proprietary control of the MAYA software search engine, and keeps its proprietary position through trade secret rather than patent protection. Therefore, we cannot discuss many of the technical aspects of the MAYA system's operation, both because of contract restrictions and because of our lack of knowledge.

EDS does state that its system employs no processing involving syntax, grammar, or semantics, nor does it employ Boolean operators except for secondary processing on selected tagged fields. A chart of the MAYA hardware architecture is available.

In late 1993, Research Publications International entered into an agreement with EDS to market two products employing the MAYA engine. One, *Patent Explorer,* gives the user on CD-ROM all software patents found by the MAYA engine - a number which very greatly exceeds the number so classified by the USPTO. This paper focuses on the online service which directly employs the MAYA engine.

Patent Analyzer

As the organizer of these sessions noted, it is hard for these presentations to avoid becoming a marketing show for our products, because it is only as they are embodied in products that the concepts of non-Boolean searching are moving from the lab to the mainstream. So, from ad text:

"MAYA goes beyond traditional Boolean or key word searching to analyze the entire structure of fulltext documents and queries employed as searches.

The most striking feature of the Patent Analyzer is that a complex document can be used as a search query, without any special user search training being required. So another patent, or a legal opinion, or an engineering document, can be used to represent the user's search - no big worry about the 'right search term' !

The results, too, are different - not just a random list of documents with one or two words matching, but a ranked, ordered list of the patents with closest match to the user's query. Never a "no-hit" or an overly long list with little relevance to the user.

The unique nature of the MAYA concept search - performed on a massively parallel supercomputer complex at EDS headquarters in Dallas - is that *the costs of the search can be predicted before the searcher puts in the query.* .

Accordingly, RP is bundling Patent Analyzer to the user market as a fixed price custom service, not a "online service" with an open-ended bill to be paid - each search costs the same, and the published price list can be guaranteed. This is totally unlike online services found today, where the cost of the search is often an unpleasant surprise to the user.

Costs are competitive with the online services which have USPTO fulltext available - but results are better! First, the user gets an edited, ordered list of patents related to his query - with full images as an option. Second, compared to manual patent searches, Patent Analyzer delivers overnight a quality of search that a manual search firm could require weeks to do."

The product Rapid Patent is offering, then, is neither a traditional online search utility, nor a custom patent search service. It is, rather, a fully bundled package which combines the benefits of non-Boolean searching with other product components we have found necessary to produce a satisfying product for a very particular class of users.

Patent Analyzer has three components: the MAYA engine, the professional editorial management, and the product presentation.

I he Analyzer MAYA Search Process

It is important to recognize that MAYA and the Patent Analyzer application are at the present *not end-user systems*. The manner in which the system operates is as a staff mediated system. We manage a "client" Sun workstation at our Arlington offices, where searches are prepared on behalf of our clients - sometimes onsite patent searchers, since we are in the complex where the USPTO public reading room resides, but as often for remote clients who have phoned in a request, faxed in a search statement, or sent as ASCII document file to use as the search query to MAYA.

The search may be as simple as a patent number, an asignee, or a key word phrase. It may, more interestingly, be a document of any length [including, confusingly, a patent number which then uses *the text of that patent* as the search query]. Various options, tailored to the needs of patent searching, are available. The client station provides many sophisticated functions, but the primary USPTO database (about 70gb in size) resides at EDS' R&D lab in Plano, Texas. Backup image files (CCITT G3/4) are at Rapid Patent in Arlington on a CD-ROM ten-jukebox network.

Establishing the search is managed from a series of X-Windows programs and DOS windows applications programs. We have a direct dedicated line to EDS' MAYA center.

Results are received from EDS and then formatted on a word-processing program to provide very readable Report output (an example is available upon request).

Most importantly, the Report does not just provide "hits" - it provides a ranked, numeric evaluation of the relation between the query and the **full content** of the patents provided as a result of the search. While the advantatages of this approach are not new to this audience - no failures and not excessive hits - this approach to the Report product is a major factor in user satisfaction.

Searching and Editing

Having been aware of the problems of some of the pioneering non-Boolean, "natural language" systems and the hybrids built on Boolean structures but presenting less structured user interfaces, Rapid Patent wanted to present a 'fail-safe' product to its particular users.

The present operating mode of the MAYA system differs from traditional online search algorithms which utilize pre-structured index files for the initial level of search. In MAYA, the entire search is performed against the (USPTO) database at the time of request. A search typically takes 3-4 minutes to return a response, but that response is a completed "Patent Report", not a mere 'hit' listing.

The result, though, is that MAYA cannot be presented to the typical online user as the same kind of "online" system which is the present industry standard: one does not just create a series of inquiries, looking for the response of the computer, in order to then obtain a successful search result. [Of course, it is just this requirement of a current online system which makes it almost useless to all but the trained online searcher.]

Rapid Patent, therefore, has bundled MAYA searching with a critical system element - an experienced human editor. Both before and after the MAYA search is executed, a patent search expert [*<u>not</u> an online search expert, but a staff professional patent researcher*] reviews the client's search to optimize the results of the MAYA retrieval.

We have found that the MAYA "concept engine", as EDS characterizes it, has many very powerful features - but also has certain differences from manual patent searching and from online search systems. It is most effective at this time to deal with these by the intervention of human expertise, and we believe this will be the case for some time to come in this particular online market.

Product Presentation

Because Rapid Patent is positioning a search product which is the most complex and expensive of our product line, and our users are dealing with information of very great value to their corporations or law firms, we are taking pains to present the product in a way which fully presents the value of the information it contains.

Thus, the text - citations, abstracts, or fulltext - is presented in readable typeset format from a high-quality laserprinter. In addition, Rapid Patent offers the highest quality image copies [many in digital form from a network jukebox CD-ROM digital image collection] if the ASCII fulltext is not sufficient.

Since many of the *Patent Analyzer* reports are of considerable size, we are offering file-cabinet packaging of print reports, and the provision of digital editions of each report, including custom CD-ROM discs.

Summary

What Rapid Patent has at hand is a quite successful product introduction - but also a case study in the reality that advanced search systems (Boolean or not) are "not quite ready for prime time". We believe very considerable investments in increased expertise will be required before these tools will be ready for the dual audiences before us in the decade entering the 21st century:

>the very sophisticated user, represented by our present patent users; &

>the totally untrained user, those who may have some success with a small CD-ROM product but none with files the size of those with which we are working.

Therefore we expect meetings like this one will be of increased value as we all gain real working, product experience with systems which do not have to explain AND and OR; or is it AND or OR ?

For further information, contact:

Michael W. Toohey, President

Rapid Patent, 1921 Jefferson Davis Highway, Suite 1821 D, Arlington, VA 22202 800-336-5010

Brett Butler, President

INFOUR, 810 Andromeda Lane, Foster City, CA 94404 415-286-1760

MAYA™

CONCEPT SEARCH MACHINE

THE TECHNOLOGY BEHIND CONCEPT SEARCHES

THE MAYA CONCEPT SEARCH MACHINE

The Maya Concept Search Machine is an EDS-designed, high performance concept search engine. MAYA has a massive main memory (up to 128 gigabytes) which is accessed by multiple processors executing in parallel to perform searches. MAYA also has advanced software that uses the large-grain data flow graph method to represent tables and queries.

THE MAYA ADVANTAGE

With its huge memory, MAYA has a distinct advantage over other concept search engines because the others use disk devices for storage instead of main memory. These relatively slow disk devices alter design decisions and force implementors to focus on optimizing access to the data stored on disk.

MAYA's combination of advanced hardware and software maximizes fast access to its main memory and is vastly superior to concept search engines that store data on disks. Also consider that MAYA is accessing its main memory through parallel multiple processors which makes response time even quicker.

MAYA's custom hardware environment and the advanced principles used in designing the software are the main reasons the MAYA machine outperforms other concept search engines.

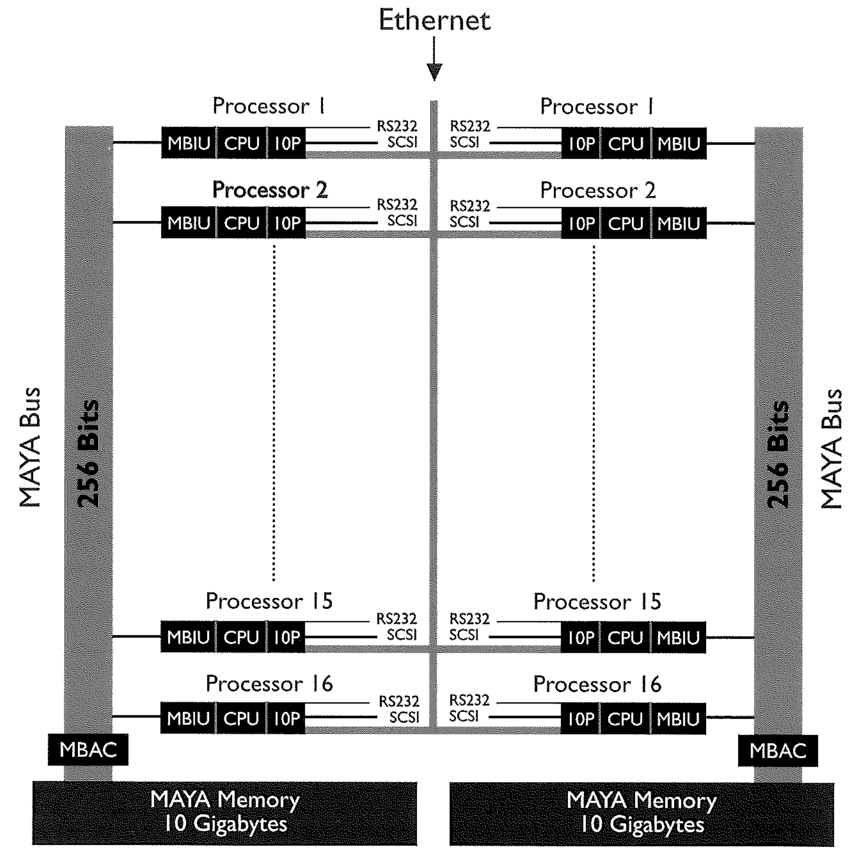
WORLDWIDE PATENT
RAPID PATENT
INFORMATION

SPO™

ADVANCED SEARCH TECHNIQUES

Our state-of-the-art technology enables us to use search methods not previously available and apply them to an 80 gigabyte database. EDS-SPO has developed a proprietary statistical linguistics approach for valuing the words in patents and other textual documents. For example, our technology can analyze the text of different documents and identify those documents that deal with similar subject matter. Applying this to patents, we can compare patents to patents or non-patents to patents to determine similarities in the concepts and ideas expressed in the documents.

Extensive testing proves that the EDS-SPO statistical linguistics approach yields much better results than key word searches and most manual searches performed by other patent search services. Consequently, we can offer customers a fast, accurate, and cost effective paent search service. No other patent search service can match our results because they can't match our technology.



The above diagram depicts the MAYA hardware with a 32-processor configuration. The diagram shows:

- An Ethernet connection, through which users send queries to and receive answers from MAYA.
- An RS232 port for sending messages to the operator console.
- A Small Computer System Interface (SCSI) peripheral bus for connecting MAYA to tape devices for logging.
- An Input/Output Processor (IOP) that serves as the interface between MAYA and outside world.
- A Reduced Instruction Set Computer (RISC) CPU board with additional local memory.
- A MAYA Bus Interface Unit (MBIU) for access to the massive main memory.
- A 256-bit bus, through which multiple processors access memory.
- A MAYA Bus Arbiter and Controller (MBAC) for scheduling I/O access to the MAYA bus via the MIBUs.
- A massive main memory configured to 20 gigabytes but with a potential of 128 gigabytes.

Rapid Patent
1921 Jefferson Davis Highway. • Suite 1821 D • Dept. IC
P.O. Box 2527 • Eads Station
Arlington, VA 22202

RAPID PATENT
1·800·336·5010

EDS-SPO
1·800·258·6739

WORLDWIDE PATENT
RAPID PATENT
INFORMATION

PATENT ANALYZER
MAYA Search Service

ASIDIC Spring 1994 Meeting
Albuquerque, NM

Sample Search

□ Listing of Patent # 4,870,568

Method for Searching a Database System Including Parallel Processors

Brewster Kahle and Craig Stanfill
Thinking Machines Corporation

■ Infringement Search Report

Employing Patent # 4870568 as the search argument

PATENT LISTING
4,870,568

Rapid Patent
1921 Jefferson Davis Highway, Suite 1821D
Arlington, Virginia 22202
1-800-336-5010

March 15, 1994

United States Patent

Patent Number: **4,870,568**

Date of Patent: **19890926**

Method for searching a database system including parallel processors

Inventor(s):

Kahle; Brewster Boston, MA USX
Stanfill; Craig W. Belmont, MA USX

Assignee(s):

Thinking Machines Corporation, Cambridge, MA

Appl. No: **878532**
Filed: **19860625**

Related U.S. Application Data

Foreign Application Priority Data

U.S. Cl. **364/200**
364/2229
364/2253
364/2254
364/2531
364/2557

Int. Cl. .... **G 06F 7/00**
Field of Search .... 364/200 MS File;200;900;300

References Cited

UNITED STATES PATENTS

04118788	19781000	Roberts	364/900 N
04152762	19790500	Bird et al.	364/200 N
04255796	19810300	Gabbe et al.	364/900 N
04358824	19821100	Glickman et al.	364/200 N
04358824	19821100	Glickman et al.	364/200 N
04451901	19840500	Wolfe et al.	364/900 N
04464650	19840800	Eastman et al.	364/200 N
04468728	19840800	Wang	364/200 N
04495566	19850100	Dickinson et al.	364/200 N
04543630	19850900	Neches	364/200 N
04554631	19851100	Reddington	364/300 N

FOREIGN PATENTS OR APPLICATIONS

OTHER PUBLICATIONS

Primary Examiner - Shaw; Gareth D.
Assistant Examiner - Ruiz; Adolfo
Attorney(s), Agent(s), or Firm(s) -
Pennie & Edmonds

ABSTRACT

A method to operate on a single instruction multiple data (SIMD) computer for searching for relevant documents in a database which makes it possible to perform thousands of operations in parallel. The words of each document are stored by surrogate coding in tables in one or more of the processors of the SIMD computer. To determine which documents of the database contain a word that is the subject of a query, a query is broadcast from a central computer to all the processors and the query operations are simultaneously performed on the documents stored in each processor. The results of the query are then returned to the central computer. After all the search words have been broadcast to the processors and point values accumulated as appropriate, the point values associated with each document are reported to the central computer. The documents with the largest point values are then ascertained and their identification is provided to the user.

15 Claims, 4 Drawings, 4 Figures

Method for searching a database system including parallel processors

N

CROSS REFERENCE TO RELATED PATENTS AND APPLICATIONS

Related applications are "Parallel Processor", Ser. No. 499,474 filed May 31, 1983, now U.S. Pat. No. 4,598,450, "Method and Apparatus for Interconnecting Processors in a Hyper-Dimensional Array", Ser. No. 740,943, filed May 31, 1985, now U.S. Pat. No. 4,805,091, and "Method of Simulating Additional Processing in a SIMD Parallel Processor Array", Ser. No. 832,913, filed Feb. 24, 1986, now U.S. Pat. No. 4,773,038, all of which are incorporated herein by reference. Related patents are "Parallel Processor/ Memory Circuit", U.S. Pat. No. 4,709,327 and "Method and Apparatus for Routing Message Packets", U.S. Pat. No. 4,598,400, both of which are incorporated by reference. BACKGROUND OF THE INVENTION

This relates to the searching of large databases and in particular to the searching of large databases in which the search strategies are executed in parallel.

Today it has become increasingly popular to store information such as articles from newswires and newspapers, abstracts and articles from journals and other print media, encyclopedias and bibliographies, on large databases for computerized search and retrieval. For convenience of reference, each group of related information will be referred to as a document regardless of its format or original physical embodiment. The methods used in searching large databases have been limited by the sequential computers available to perform the search. Ideally a search method should have a high recall and precision. Recall is the proportion of relevant documents in the entire database which are retrieved. Precision is the proportion of retrieved documents which are relevant. Exhaustive search methods provide high recall and precision. The basic problem is that an exhaustive search may take a very long time. Therefore, non-exhaustive methods are used.

The usual method of organizing a database is a technique called "inverting the database". See G. James, Document Databases (Van Nostrand Reinhold Company 1985); C. J. Rijsbergen, Information Retrieval, p. 72 (Butterworths, 2d ed. 1979). Each document is assigned a unique document number. The words in the documents (excluding trivial words such as "a" and "the") are tagged with the document number and placed in an alphabetical index. To locate all documents containing a given word, the index is searched for that word, and a set of document numbers is returned. Alternatively, the words of each document may be stored by surrogate coding in which each word is represented by a hash code in a table of hash codes and a word search is performed by searching for the presence of the hash code associated with the word in interest.

To search for documents containing more than one word, a boolean search strategy is typically used on the inverted index. A boolean search is a search which achieves its results by logical comparisons of the query with the documents. Commercial application of this technique requires rooms full of disk drives and large mainframe computers. The response time is often quite slow depending on the complexity of the query because the search through the index and the logical comparisons are executed sequentially. Such systems are limited in the quality of the search they provide and are found to be clumsy to use. There is a tradeoff between recall and precision which limits the quality of boolean searches on large databases. Searching a database for documents containing a single word may lead to low recall, because there is no guarantee that all relevant documents will use that word. In addition, it is likely that a large number of irrelevant documents will be retrieved, leading to low precision. Searching for several words aggravates these problems. If the searcher looks for any of several words (a disjunctive query), recall improves but precision goes down. If the searcher looks for documents containing all of several words (a conjunctive query), precision improves but recall suffers. For a large database, this means that the searcher may have to choose between missing important information or searching through thousands of irrelevant documents. There are additional problems with the viability of using boolean queries for full text search. First, the user is playing a guessing game, trying to guess which words the authors of the documents he is interested in might have used. Second, even if he guesses the words, he has to figure out which connectives to use to avoid getting too much or too little data. This often involves several iterations as the user debugs his query. Finally, the syntax of boolean queries is complex, making the system difficult to learn.

A second search strategy employs a variant on boolean queries referred to as "simple queries". See C. J. van Rijsbergen, Information Retrieval, p. 160 (Butterworths, 2d ed. 1979). In this search strategy a query consists of a set of words, each of which is assigned a point value. Every document in the database is scored by adding up the point values for the words it contains. The result of this query is a set of documents, ordered by their total point values. Simple queries are comparable to boolean queries in the quality of the search they support. For example, if the user looks only at the documents which have a positive score, he is essentially looking at the results of a disjunctive query, and can expect high recall but low precision. An advantage of simple queries is that, between these two extremes, there are regions of intermediate recall and precision. In addition, they are easier to use than boolean queries. The user does not need to decide which connectives to use as there are none. The user does not need to learn a complex query language, as the query consists of a list of words. However, searching with simple queries, like searching with boolean queries, remains a guessing game. An additional problem is determining where to set the threshold in the point value of responses from the query in order to limit the number of retrieved documents to a manageable amount.

Another search strategy is relevance feedback. In this strategy simple queries are constructed from the texts of documents judged to be relevant. See G. Salton, The SMART Retrieval System-Experiment in Automatic Document Processing, p. 313 (Prentice-Hall 1971); C. J. van Rijsbergen, Information Retrieval, p. 105 (Butterworths, 2d ed. 1979). First, a search method is used to locate a small set of possibly relevant documents. The user then scans these documents, and marks any which he considers obviously relevant as good and any which he considers obviously irrelevant as bad. The text of the marked documents is then scanned for appropriate search words, and a query is constructed from these words. The more good documents a word occurs in, the greater its importance in the new query and therefore the higher the score assigned to that word. The new query may contain hundreds of terms. This query is then applied to the database in the same fashion as a simple query. Relevance feedback leads to both high precision and high recall due to the large number of words employed in the search process. One word taken by itself conveys little information; but several hundred words together convey a great deal. Only highly relevant documents will use a high proportion of this set of several hundred items. However, the only way to implement such a query is by an exhaustive search which is impracticable on the serial mainframe systems currently in use for database retrieval systems. SUMMARY OF THE INVENTION

The present invention relates to the use of a massively parallel processor for document search and retrieval. The system as presented is sufficiently fast to permit the application of exhaustive search methods not previously feasible for large databases.

In the preferred embodiment of the invention the document data is stored and the searches are implemented on a single instruction multiple data (SIMD) computer which makes it possible to perform thousands of operations in parallel. One such SIMD computer on which the invention has been performed is the Connection Machine (Reg. TM) Computer made by the present assignee, Thinking Machines, Inc. of Cambridge, Massachusetts. This computer is described more fully in U.S. Pat. No. 4,598,400, which is incorporated herein by reference. In the embodiment of the Connection Machine Computer on which the invention has been practiced, the computer comprises 65,536 relatively small identical processors which are interconnected in a sixteen-dimensional hypercube network.

The words of each document are stored by surrogate coding in tables in one or more of the processors of the SIMD computer. To determine which documents of the database contain a word that is the subject of a query, a query is broadcast from a central computer to all the processors and the query operations are simultaneously performed on the documents stored in each processor. The results of the query are then returned to the central computer.

Because of the enormous parallel processing capability of an SIMD computer such as the Connection Machine Computer, simple query and relevance feedback search strategies using large numbers of search words and exhaustive or near exhaustive search strategies are now practical. Scoring of the results of such searches is done in parallel at each processor. For example, each processor which stores a hash value associated with a word that is the subject of a query can be directed to accumulate a point value for that word. After all the search words have been broadcast to the processors and point values accumulated as appropriate, the point values associated with each document are reported to the central computer. The documents with the largest point values are then ascertained and their identification is provided to the user.

DESCRIPTION OF THE DRAWINGS

These and other objects, features and elements of the invention will be more readily apparent from the following Description of the Preferred Embodiment of the Invention in which:

FIGS. 1 and 2 depict in schematic form details of a SIMD processor preferably used in the practice of the invention;

FIG. 3 is a block diagram of the search and retrieval process; and

FIG. 4 is a block diagram of the query forming process.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

In the system of the invention, a single instruction multiple data (SIMD) computer such as the Connection Machine Computer is preferably used. This computer is described in detail in U.S. Pat. No. 4,598,400.

As shown in FIG. 1A of that patent which is reproduced in FIG. 1, the computer system comprises a central computer 10, a microcontroller 20, an array 30 of parallel processing integrated circuits 35, a data source 40, a first buffer and multiplexer/demultiplexer 50, first, second, third and fourth bidirectional bus control circuits 60, 65, 70, 75, a second buffer and multiplexer/demultiplexer 80, and a data sink 90. Central computer 10 may be a suitably programmed commercially available computer such as a Symbolics 3600-series LISP Machine. The database to be searched is stored as described below in the memories of individual processor/memories 36 in integrated circuits 35.

Microcontroller 20 is an instruction sequencer of conventional design for generating a sequence of instructions that are applied to array 30 by means of a thirty-two bit parallel bus 22. Microcontroller 20 receives from array 30 a signal on line 26. This signal is a general purpose or GLOBAL signal that can be used for data output and status information. Bus 22 and line 26 are connected in parallel to each IC 35. As a result, signals from microcontroller 20 are applied simultaneously to each IC 35 in array 30 and the signal applied to microcontroller 20 on line 26 is formed by combining the signal outputs from all of ICs 35 of the array.

In the embodiment of the Connection Machine Computer used in the practice of the present invention, array 30 contains 4096 (=212) identical ICs 35; and each IC 35 contains 16 (=24) identical processor/memories 36. Thus, the entire array 30 contains 65,536 (=216) identical processor/memories 36.

Processor/memories 36 are organized and interconnected in two geometries: a conventional two-dimensional grid pattern and a 16 dimension hypercube network. In the grid pattern, the processor/memories are organized in a rectangular array and connected to their four nearest neighbors in the array. The sides of this array and the four neighbors are identified as NORTH, EAST, SOUTH and WEST. The hypercube network allows processors to communicate by exchanging packets of information. This configuration is realized by organizing and interconnecting the IC's 35 in the form of a Boolean n-cube of sixteen dimensions. Each IC 35 is provided with logic circuitry to control the routing of messages through such an interconnection network; and within each IC, bus connections are provided to every processor/memory so that each processor/memory can communicate with any other by sending a signal through at most sixteen communication lines.

An illustrative processor/memory 36 is disclosed in greater detail in FIG. 2 which is the same as FIG. 7A of U.S. Pat. No. 4,598,400. As shown in FIG. 2, the processor/memory comprises random access memory (RAM) 250, arithmetic logic unit (ALU) 280 and flag controller 290. The ALU operates on data from three sources, two registers in the RAM and one flag input, and produces two outputs, a sum output that is written into one of the RAM registers and a carry output that is made available to certain registers in the flag controller as well as to certain other processor/memories.

The inputs to RAM 250 are busses 152, 154, 156, 158, a sum output line 285 from ALU 280, the message packet input line 122 from communication interface unit (CIU) 180 of FIG. 6B of U.S. Pat. No. 4,598,400 and a WRITE ENABLE line 298 from flag controller 290. The outputs from RAM 250 are lines 256, 257. The signals on lines 256, 257 are obtained from the same column of two different registers in RAM 250, one of which is designated Register A and the other Register B. Busses 152, 154, 156, 158 address these registers and the columns therein in accordance with the instruction words from microcontroller 20. Illustratively, RAM 250 has a memory capacity of 4096 bits.

Flag controller 290 is an array of eight one-bit D-type flip-flops 292, a two-out-of-sixteen selector 294 and some logic gates. The inputs to flip-flops 292 are a carry output signal from ALU 280, a WRITE ENABLE signal on line 298 from selector 294, and the eight lines of bus 172 from programmable logic array (PLA) 150 of FIG. 6B of U.S. Pat. No. 4,598,400. Lines 172 are address lines each of which is connected to a different one of flip-flops 292 to select the one flip-flop into which a flag bit is to be written. The outputs of flip-flops 292 are applied to selector 294.

The inputs to selector 294 are up to sixteen flag signal lines 295, eight of which are from flip-flops 292, and the sixteen lines each of busses 174, 176. Again, lines 174 and 176 are address lines which select one of the flag signal lines for output or further processing. Selector 294 provides outputs on lines 296 and 297 that are whichever flags have been selected by address lines 174 and 176, respectively.

ALU 280 comprises a one-out-of-eight decoder 282, a sum output selector 284 and a carry output selector 286. As detailed in U.S. Pat. No. 4,598,400, this enables it to produce sum and carry outputs for many functions including ADD, logical OR and logical AND. ALU 280 operates on three bits at a time, two on lines 256, 257 from Registers A and B in RAM 250 and one on line 296 from flag controller 290. The ALU has two outputs: a sum on line 285 that is written into Register A of RAM 250 and a carry on line 287 that may be written into a flag register 292 and applied to the inputs of the other processor/memories 36 to which this processor/memory is connected.

The words of the document are stored in a table format originally developed for spelling correction dictionaries called surrogate coding. See Dodds, "Reducing Dictionary Size by Using a Hashing Technique", Communications of the ACM, Vol. 25, No. 6, pp. 368-370, (June 1982); Nix, "Experience With a Space Efficient Way to Store a Dictionary", Communications of the ACM, Vol. 24, No. 5, pp. 297-298, (May, 1981); Peterson, "Computer Programs for Detecting and Correcting Spelling Errors", Communications of the ACM, Vol. 23, No. 12, pp. 676-687. (December, 1980). Although the tables may be of any size, it is preferred that the table be 512 or 1024 bits long. The 4096 bits of RAM allows for six tables of 512 bits or three tables of 1024 bits with the remaining memory used as scratch memory.

To store the words of a document in the table, the table is first initialized to zero. A fixed number of independent hash codes-ten, in the presently preferred embodiment-are generated for each significant word in the document. Each code corresponds to a position in the table. For example, for a table of 512 bits, each code would be between zero and 511. For each of the hash codes generated for a word, the corresponding binary bit at that address in the table is set to one. To minimize data storage requirements, trivial words such as "a" and "the" are not included in the table. In addition, a text indexer preferably is used which picks out noun phrases in a document to input into the table. This allows for a three to one compression of the document.

Each document of the database is stored in this fashion in one or more tables in one or more processor/memories of the SIMD computer. If a document contains more than the maximum number of words that should be stored in a table, additional tables are used. For example if a ninety word document is stored in the database, and thirty words are contained in each table, a total of three tables are used. The set of tables which contain a single document is called a chain. Preferably each of the tables in a chain is located in a physically different processor, and each table is located in the same portion of its respective memory. Alternatively, all the tables could be located in the same physical processor.

To probe for the presence of a word in the documents stored in the tables of the processor/memories, the corresponding bits of the ten hash codes for that word are checked in each table of the processor/memories. If any of the ten bits in a table are zero, the word is definitely absent from that table, yielding a negative response. If all ten bits are one, then the word is probably present yielding a positive response. Although this algorithm does not generate false negatives, there is a possibility of false positives. The probability in this algorithm of a false positive is dependent on the number of words contained in each table as well as the size of the table and the number of bits which are set for each word. In a table of 512 bits with ten bits set for each word, the probability of a false positive can be shown to be about one in a million for a table containing fifteen words, one in a hundred thousand for a table of twenty words and thirty in a hundred thousand for a table of thirty words. For optimal system performance, it seems preferable to limit the table to about fifteen to thirty words.

FIGS. 3 and 4 illustrates the exhaustive search strategy used in the system of the invention. To permit user access to the documents from a computer terminal, the documents are stored in full text in the central computer and a display terminal is provided for accessing this database. As indicated in Box 500 of FIG. 3, the significant words of each document are then hash coded and the hash codes are then stored in one or more tables in the memories of the processor/memories of the SIMD computer. As indicated above, trivial words are ignored and storage typically is limited to noun phrases. To begin a search, the user selects at least one word and preferably several that delineate the subject of interest. This word or words constitutes a query (Box 501). When first executing a query as indicated by the lefthand side of FIG. 4, the user enters these words into the central computer and may also assign point values to each word reflecting his estimate of the significance of the word in the search (Box 511, FIG. 4). The central computer then examines the full text of the documents stored in the central computer for the presence of one or more of those words (Box 502, FIG. 3), computes point value scores for the documents (Box 503, FIG. 3). if applicable, and identifies to the user the documents selected by this process (Box 504, FIG. 3). As indicated by the righthand side of FIG. 4, the user then examines the documents and informs the central computer which documents are relevant or "good" and which are irrelevant or "bad"(Box 521, FIG. 4). The computer then examines these documents to locate appropriate search words and formulates a query from these words. As indicated in Box 523, the computer also assigns to these words a point value basing its valuation, for example, on the number of good documents in which the word appears. Other parameters, such as the frequency of occurrence of a particular word for example, words which do not appear frequently in a document would have a higher point value assigned to the word, whether a word occurs in the title or headlines, and whether the user has made an explicit note of the word, may also be used in constructing the simple query (Box 524, FIG. 4). As indicated by Box 525, the resulting query may contain hundreds of words.

As indicated in Box 502, FIG. 3, this query is then used to search the hash tables stored in the memories of the SIMD computer. For each word in the query, the central computer determines from a look-up table the values or addresses in the hash table where the ten bits of the corresponding hash code are stored. It then instructs each processor/memory to read the bit at each of those addresses. Each bit that is read is used to set a flag and this flag is then ANDed together with the next bit that is read to determine the next value of the flag. If any of the hash code bits has a value of zero, the flag becomes zero, and the test fails, indicating that the word in question is not stored in that table. If the test succeeds, the flag is a one, the word is assumed to be in the document represented by that table and, as indicated by Box 503 of FIG. 3, the point value associated with that word is awarded to that document and accumulated with any other point values associated with other words in the document.

Advantageously, communications from the individual processor/memories to the central computer can be minimized by storing the point values for each document at the individual processor/memories until completion of the entire query search. Point values are accumulated after each word is tested by broadcasting to all the processor/memories an instruction to accumulate the point value if the flag bit is one.

If a document is divided among multiple tables, the values in each table are passed to the first table in the chain where they are accumulated.

Documents having the largest point values are then identified by sorting the point values stored in the processor/memories in order of magnitude (Box 504, FIG. 3). Computer programs for such a sort are well known in the art and their adaptation to a SIMD computer will be apparent from the foregoing description. Alternatively, the point values can be tested to identify the maximum point values and the identification of the documents associated with a series of such maxima can be extracted from the processor/memories. To perform such a test, the central computer simultaneously tests the most significant bit of the point values stored in each of the processor/memories. This is readily done if the point values are all stored at the same addresses in all the processor/memories. The test is in the form of an instruction to set a first flag and ignore further parts of the test if the most significant bit is zero and to produce an output on the GLOBAL signal line 26 of FIG. 1 if the most significant bit is a one. If no output is received on line 26 from any processor/memory, the central computer resets all the flags and their processor/memories and begins the test anew with the next most significant bit. If an output is received, the central processor enters the next cycle of the test and issues an instruction to those processor/memories still being tested to set a second flag and ignore further parts of the test if the test of the next most significant bit is zero and to produce an output on the GLOBAL signal line if that bit is a one. If no output is received on the GLOBAL line, the second flags of the processor/memories that were shut down during that cycle of the test are reset, those processor/memories are reactivated and the third most significant bit is tested. If, however, an output is received, the first flags are set in those processor/memories whose second flags were set; and the central processor enters the next cycle of the test. It then tests the third most significant bit and so on. As a result of this process, the processor/memory having the maximum point value is isolated in the array and the document associated with that point value is identified. The point value associated with that document is then set to zero and the process is repeated to find the document having the next highest point value; and so on.

Since a large number of documents may be retrieved using the simple query search strategy, it is preferred that some means be used to limit the number of retrieved documents to a managable amount. Preferably, this is done by retrieving the best documents (i.e., those with the highest point values) first and stopping when enough documents have been retrieved. Alternatively, a threshold point value can be established so that if the total point value of the document is below the threshold point value, the document is not retrieved.

It has been found that by utilizing the system of the invention the user can obtain both high recall and precision. Exhaustive searches which were not practicable can now be used. Since the searches are executed in parallel the search time is extremely fast. For example, a simple query search of 200 terms on a 112 Megabyte database is executed in 60 milliseconds.

As will be apparent to those skilled in the art, numerous modifications may be made within the scope of the above described invention. While the invention has been described in terms of parallel processing implementation of a combination of simple queries and relevant feedback search strategies other search strategies such as boolean search strategies and other exhaustive search strategies may be used.

What is claimed is:

1. A process for searching for relevant documents in a database comprising the steps of:

(a) forming a database by storing for each of a plurality of documents at least one table of hash codes representing words in the document, the table(s) that represent the words in each different document being stored in a different digital data processor, each hash code comprising information at a plurality of bit locations;

(b) forming a query having at least one word and a point value of relevance assigned to each word;

(c) testing if the word in the query is in the database by:

(1) determining the bit locations in the table at which the hash code corresponding to the queried word is stored; and

(2) simultaneously testing in each of the processors the bit locations corresponding to the queried word;

(d) adding at each digital data processor the point value associated with the queried word to a total point value for the document if the hash code is found at all the bit locations corresponding to the queried word that are tested in that processor; and

(e) providing identification of those documents in the database with high total point values.

2. The process of claim 1 wherein the step of forming a query comprises:

(a) identifying a group of relevant documents;

(b) determining a frequency at which the words in the documents occur among all the relevant documents; and

(c) generating a query based on the frequency of occurrence of words in the relevant documents.

3. The process of claim 1 wherein the step of adding the point value of a queried word to the total point value of a document comprises:

(a) setting a flag in the processor to indicate the presence of the queried word in the table;

(b) communicating simultaneously from a central computer to each of the processors a command to add the point value assigned to the word to the total point value of the document if the flag in the processor is set.

4. The process according to claim 1 wherein a document is represented by a chain of tables, each of which stores a representation of a portion of the document, each chain of tables comprising a first table and one or more successive tables, the step of adding the point value associated with the queried word to the total point value of the document comprising the steps of:

(a) communicating the total point value of the portion of the document represented by each successive table in the chain to the preceding table in the chain; and

(b) accumulating the total point values of all the tables in the first table of the chain.

5. The process according to claim 1 wherein the storing of document data in the processors comprises the steps of:

(a) initializing to zero a table of bits in a memory of each processor;

(b) generating a plurality of independent hash codes for each word with values in an address range of the table; and

(c) for each hash code, setting a bit at an address in the table corresponding to each hash code value.

6. The process of claim 1 wherein the step of providing the identification of those documents with high total point values comprises identifying the documents in numeric order of total point value, said process comprising:

(a) testing the most significant bit of the point value stored in memory in each processor;

(b) setting a flag in the processor if the most significant bit is zero;

(c) repeating step (a) and (b), if necessary, until a non-zero bit is identified; and

(d) successively testing bits of the point values of lesser significance in those processors where a non-zero bit is identified until a document is identified with the highest total point value.

7. The process of claim 1 wherein the step of providing the identification of those documents with high total point values comprises identifying the documents in numeric order of total point value, said process comprising:

(a) testing the most significant bit of the point values stored in memory in each processor;

(b) setting a flag in the processor if the most significant bit is zero;

(c) testing the next most significant bit of the point values stored in memory at each processor where a flag is not set; and

(d) setting a second flag in the processor if the next most significant bit is zero.

8. The process of claim 1 wherein the step of providing the identification of documents with high total point values comprises:

(a) successively testing at each digital data processor bits of the total point values associated with each document beginning with the most significant bit and continuing to test bits of lesser significance until the highest total point value is identified; and

(b) repeating step (a) of this claim on the total point values remaining after the highest total point value is identified in the previous execution of step (a).

9. The method of claim 1 wherein different point values are assigned to at least some different words.

10. A process for searching in a database comprising the steps of:

(a) forming a database by storing in each of a plurality of digital data processors at least one table of hash codes, the hash codes in each table representing a group of related words;

(b) forming a query having at least one word and a point value of relevance assigned to each word;

(c) testing if the word in the query is in the database by:

(1) determining the bit locations in the table at which the hash code corresponding to the queried word is stored; and

(2) simultaneously testing in each of the processors the bit locations corresponding to the queried word;

(d) at each digital data processor, adding the point value associated with the queried word to a total point value for that group of related words if the hash code is found at all the bit locations tested in the table; and

(e) providing identification of those groups of related words in the database with high total point values.

11. The process of claim 10 wherein the step of adding the point value of a queried word to the total point value for a group of related words comprises:

(a) setting a flag in the processor to indicate the presence of the queried word in the table;

(b) communicating simultaneously from a central computer to each of the processors a command to add the point value of the queried word to the total point value for the group of related words if the flag in the processor is set.

12. The process of claim 10 wherein the step of providing the identification of those groups of related words with high total point values comprises:

(a) successively testing the bits of the total point values associated with each group of related words beginning with the most significant bit and continuing to test bits of lesser significance until the highest total point value is identified; and

(b) repeating step (a) of this claim on the total point values remaining after the highest total point values is identified in the previous execution of step (a).

13. The process of claim 10 wherein the step of providing the identification of those groups of related words with high total point values comprises identifying the groups in numeric order of total point value, said process comprising:

(a) testing the most significant bit of the point values stored in memory in each processor;

(b) setting a flag in the processor if the most significant bit is zero;

(c) testing the next most significant bit of the point values stored in memory at each processor where a flag is not set;

(d) setting a second flag in the processor if the next most significant bit is zero.

14. The method of claim 10 wherein different point values are assigned to at least some different words.

15. A process for searching in a database comprising the steps of:

(a) forming a database by storing in each of a plurality of digital data processors at least one table of hash codes, the hash codes in each table representing a group of related words;

(b) forming a query having at least one word;

(c) testing for the presence of the queried word in the database by:

(1) determining the bit locations in the table at which the hash code corresponding to the queried word is stored; and

(2) simultaneously testing in each of the processors the bit locations corresponding to the queried word; and

(d) scoring each group of related words, a score for a group of related words being increased if the hash code is found at all the bit locations tested in the table.

* * * * *

INFRINGEMENT SEARCH REPORT

Patent 4,870,568

Rapid Patent
1921 Jefferson Davis Highway, Suite 1821D
Arlington, Virginia 22202
1-800-336-5010

March 15, 1994

Patent Number: 4,870,568

Patent Number: 4,870,568
Class: 364 **Date Issued: 19890926** **Distance: 0.999999**
SubClass: 200 **Date Filed: 19860625**
Inventors: Kahle; Brewster:Stanfill; Craig W.
Assignee: Thinking Machines Corporation

Method for searching a database system including parallel processors

A method to operate on a single instruction multiple data (SIMD) computer for searching for relevant documents in a database which makes it possible to perform thousands of operations in parallel. The words of each document are stored by surrogate coding in tables in one or more of the processors of the SIMD computer. To determine which documents of the database contain a word that is the subject of a query, a query is broadcast from a central computer to all the processors and the query operations are simultaneously performed on the documents stored in each processor. The results of the query are then returned to the central computer. After all the search words have been broadcast to the processors and point values accumulated as appropriate, the point values associated with each document are reported to the central computer. The documents with the largest point values are then ascertained and their identification is provided to the user.

Patent Number: 4,849,898
Class: 364 **Date Issued: 19890718** **Distance: 0.414106**
SubClass: 419 **Date Filed: 19880518**
Inventors: Adi; Tammam
Assignee: Management Information Technologies, Inc.

Method and apparatus to identify the relation of meaning between words in text expressions

A text comprehension and retrieval method and apparatus that uses letter-semantic analysis of the micro-syntax, or the syntax between the letters, in two words to measure how much two words are related as to their meanings or the human language concepts they present. Letter-semantic analysis involves assigning numerical values to the letters of a first word and a second word based on the dual characteristics of orientation and category inherent in the letters, and then analyzing those numerical values to identify the semantic relatedness of the letters of the first word to the letters of the second word. A letter semantic-matrix assigns weights to the meaningful letters to allow the application of letter semantic rules to convert the concepts represented by the letters of the words to numeric values. The numeric values represent the amount of relatedness of the first word to the second word and are used to retrieve text from documents having concepts related to a user supplied query expression.

Patent Number: 5,278,980
Class: 395 **Date Issued: 19940111** **Distance: 0.403893**
SubClass: 600 **Date Filed: 19910816**
Inventors: Pedersen; Jan O.:Halvorsen; Per-Kristian:Cutting; Douglass R.:Tukey; John W.:Bier; Eric A.:Bobrow; Daniel G.
Assignee: Xerox Corporation

Iterative technique for phrase query formation and an information retrieval system employing same

An information retrieval system and method are provided in which an operator inputs one or more query words which are used to determine a search key for searching through a corpus of documents, and which returns any matches between the search key and the corpus of documents as a phrase containing the word data matching the query word(s), a non-stop (content) word next adjacent to the matching word data, and all intervening stop-words between the matching word data and the next adjacent non-stop word. The operator, after reviewing one or more of the returned phrases can then use one or more of the next adjacent non-stop-words as new query words to reformulate the search key and perform a subsequent search through the document corpus. This process can be conducted iteratively, until the appropriate documents of interest are located. The additional non-stop-words from each phrase are preferably aligned with each other (e.g., by columnation) to ease viewing of the "new" content words.

Infringement Search Report

Patent Number: 4,965,763
Class: 364 Date Issued: 19901023 Distance: 0.375554
SubClass: 900 Date Filed: 19890206
Inventors: Zamora; Elena M.
Assignee: International Business Machines Corporation

Computer method for automatic extraction of commonly specified information from business correspondence

A Parametric Information Extraction (PIE) system has been developed to identify automatically commonly specified information such as author, date, recipient, address, subject statement, etc. from documents in free format. The program-generated data can be used directly or can be supplemented manually to provide automatic indexing or indexing aid, respectively.

Patent Number: 5,201,048
Class: 395 Date Issued: 19930406 Distance: 0.370054
SubClass: 600 Date Filed: 19910821
Inventors: Coulter; Erick S.:Richards; Thomas A.
Assignee: Axxess Technologies, Inc.

High speed computer system for search and retrieval of data within text and record oriented files

Disclosed is a system which provides for the high-speed search and retrieval of both text and record-oriented information from one or more files stored in a computer or computer network. The system first creates a search file with a structure that can be quickly searched for each reference or boolean combination of references using, as input, either the full data to be matched or only partial data and "wildcard" symbols. This search file can be placed on a computer different from the original data files, to facilitate searching in a network environment.

Patent Number: 5,099,426
Class: 364 Date Issued: 19920324 Distance: 0.364168
SubClass: 419 Date Filed: 19890119
Inventors: Carlgren; Richard G.:Modlin; William D.
Assignee: International Business Machines Corporation

Method for use of morphological information to cross reference keywords used for information retrieval

A data processing method is disclosed for storing and retrieving text. The storage part of the method includes the steps of compiling a vocabulary list of words occurring in the text and augmenting the vocabulary list with lemmas of the words in the text, as an augmented word list. It then continues with the steps of compiling a cross reference table relating the lemmas of the words to locations of the words in the text and storing the text, the augmented word list and the cross reference table.
The retrieval part of the method includes the steps of inputting a query word to access a portion of the stored text, searching the augmented vocabulary list using the query word as a search term, and accessing the cross reference table with a lemma of the query word to locate the portion of the stored text.
The resulting invention enables a faster performance for "fuzzy" searches of text in documents, while enabling the cross reference lists used in the retrieval process, to be compactly stored.

Patent Number: 4,774,655
Class: 364 Date Issued: 19880927 Distance: 0.362187
SubClass: 200 Date Filed: 19841024
Inventors: Kollin; Richard P.:Francis; Gerald A.:Tiano; Craig
Assignee: Telebase Systems, Inc.

System for retrieving information from a plurality of remote databases having at least two different languages

A system enables a user to retrieve information from a plurality of commercially available databases. The user gains access to the system with a personal computer equipped with a modem, or with any other terminal capable of sending and receiving data over telecommunications lines. The system presents the user with a sequence of menus which ask the user to specify an area of interest. The choices presented to the user are programmed to cover virtually the entire field of human knowledge. After the user has chosen the area of interest, the system automatically selects a database to be searched. The user then enters a specific search request, and the system translates the format of the request into a format which is appropriate for the database selected. The system establishes communication with the database, downloads the information received from the database, and terminates the link with the database. The user is then able to browse electronically through the information received, without incurring the added expense of maintaining communication with the database.

Infringement Search Report

Patent Number: **4,991,087**
Class: **364** **Date Issued:** **19910205** **Distance:** **0.356974**
SubClass: **200** **Date Filed:** **19880818**
Inventors: **Burkowski; Forbes J.:Krebs; Marke S.**
Assignee:

Method of using signature subsets for indexing a textual database

A method of operating a computer system to store and retrieve information in a database uses a signature file of the database that is divided into subsets. A word signature is mapped to a particular subset during creation of the file and the same mapping information is used to retrieve the information in response to a query word. Each word signature is a logical word signature and has two components a physical word signature and a subset designation field. In this way, when information is retrieved from the database, only that subset containing the relevant word signature is scanned. The signature file is automatically created by the system as the database is stored on the data storage modules. During retrieval, the control reviews information received from the data storage means and if a match occurs between a physical word signature for a query word and a particular physical word signature arriving from the input section, the control sends the physical word signature to the FIFO buffer in memory together with the document identifier located subsequent to the matched physical word signature. The control then moves on to process the next physical word signature received from the data storage means. If there is no match, the control ignores the physical word signature and moves on to process the next physical word signature received from the data storage means. The control is effectively capable of processing several query words in parallel.

Patent Number: **4,972,349**
Class: **364** **Date Issued:** **19901120** **Distance:** **0.347303**
SubClass: **900** **Date Filed:** **19890814**
Inventors: **Kleinberger; Paul J.**
Assignee:

Information retrieval system and method

A computerized information retriveal system is formed of a textbase of texts of variable length and content. The texts are selected from the textbase on the basis of Boolean logic searches among keywords associated with the texts. When a group is retrieved from such a search, the system automatically segregates the texts based on the presence or absence of a criterion key keyword selected so as to segregate the texts into sub-groups. The same criterion key analysis can then be applied recursively to the sub-groups. The resulting sub-groups are then displayed to the user in a hierarchical display to illustrate the relationships amoung the texts. A string comparison routine is also disclosed to search for similar keywords.

Patent Number: **5,062,074**
Class: **364** **Date Issued:** **19911029** **Distance:** **0.346262**
SubClass: **900** **Date Filed:** **19900830**
Inventors: **Kleinberger; Paul J.**
Assignee: **TNET, Inc.**

Information retrieval system and method

A computerized information retrieval system is formed of a textbase of texts of variable length and content. The texts are selected from the textbase on the basis of Boolean logic searches among keywords associated with the texts. When a group is retrieved from such a search, the system automatically segregates the texts based on the presence of absence of a criterion key keyword selected so as to segregate the texts into sub-groups. The same criterion key analysis can then be applied recursively to the sub-groups. The criterion key analysis can then be applied recursively to the sub-groups. The resulting sub-groups are then displayed to the user in a hierarchical display to illustrate the relationships among the texts. A string comparison routine is also disclosed to search for similar keywords.

Patent Number: **5,265,065**
Class: **395** **Date Issued:** **19931123** **Distance:** **0.345633**
SubClass: **600** **Date Filed:** **19911008**
Inventors: **Turtle; Howard R.**
Assignee: **West Publishing Company**

Method and apparatus for information retrieval from a database by replacing domain specific stemmed phases in a natural language to create a search query

A computer implemented process for creating a search query for an information retrieval system in which a database is provided containing a plurality of stopwords and phrases. A natural language input query defines the composition of the test of documents to be identified. Each word of the natural language input query is compared to the database in order to remove stopwords from the query. The remaining words of the input query are stemmed to their basic roots, and the sequence of stemmed words in the list is compared to phrases in the database to identify phrases in the search query. The phrases are substituted for the sequence of stemmed words from the list so that the remaining elements, namely the substituted phrases and unsubstituted stemmed words, form the search query. The completed search query elements are query nodes of a query network used to match representation nodes of a document network of an inference network. The database includes as options a topic and key database for finding numerical keys, and a synonym database for finding synonyms, both of which are employed in the query as query nodes.

Patent Number: 5,148,547
Class: 395 **Date Issued: 19920915** **Distance: 0.345428**
SubClass: 800 **Date Filed: 19910517**
Inventors: Kahle; Brewster A.:Douglas; David C.:Vasilevsky; Alexander:Christman; David P.:Yang; Shaw W.:Crouch; Kenneth W.
Assignee: Thinking Machines Corporation

Method and apparatus for interfacing bit-serial parallel processors to a coprocessor

A parallel processor is disclosed which combines the advantages of an array of bit-serial processors and an array of word-oriented processors. Further, the invention provides for ready communication between data organized in bit-serial fashion and that organized in parallel. The processor comprises a plurality of word-oriented processors, at least one transposer associated with each processor, said transposer having n bit-serial inputs and m bit parallel outputs and a bit-serial processor associated with each bit-serial input of the transposer. The parallel processor further comprises a memory for each bit-serial processor and a data bus interconnecting the memory, the bit-serial processors and the bit-serial inputs of the transposer. The transposer converts serial inputs to parallel, word organized outputs which are provided as inputs to the word-oriented processors. In accordance with a preferred embodiment of the invention, three or more transposers are used in connection with each word-oriented processor so as to provide a pipelining capability that significantly enhances processing speeds.

Patent Number: 4,860,201
Class: 364 **Date Issued: 19890822** **Distance: 0.344460**
SubClass: 200 **Date Filed: 19860902**
Inventors: Stolfo; Salvatore J.:Miranker; Daniel P.
Assignee: The Trustees of Columbia University in the City of New York

Binary tree parallel processor

A plurality of parallel processing elements are connected in a binary tree configuration, with each processing element except those in the highest and lowest levels being in communication with a single parent processing element as well as first and second (or left and right) child processing elements. Each processing element comprises a processor, a read/write or random access memory, and an input/output (I/O) device. The I/O device provides interfacing between each processing element and its parent and children processing elements so as to provide significant improvements in propagation speeds through the binary tree. The I/O device allows the presently preferred embodiment of the invention to be clocked at 12 megahertz, producing in the case of a tree of 1023 processors, each having an average instruction cycle time of 1.8 s, a system with a raw computational throughput of approximately 570 million instructions per second. The I/O device communicates data and queries from the root processing element to all other N processing elements in the array in one processor instruction cycle instead of in O(log2N) processor instruction cycles as in prior art binary tree arrays. Primitive queries are executed in parallel by each processing element and the results made available for reporting back to the root processing element. In several important cases, these results can be combined and reported back to the root processing element in a single processor instruction cycle instead of in O(log2N) processor instruction cycles as in prior art binary tree arrays. Thus, the elapsed time for a broadcast and report operation is in effect a constant time regardless of the number of processors in the array.

Patent Number: 5,220,625
Class: 382 **Date Issued: 19930615** **Distance: 0.344095**
SubClass: 54 **Date Filed: 19920717**
Inventors: Hatakeyama; Atsushi:Ando; Hiroe:Kato; Kanji:Asakawa; Satoshi:Kawaguchi; Hisamitsu
Assignee: Hitachi, Ltd.

Information search terminal and system

An information search terminal apparatus and information search system for performing information search by using a variety of windows assure high manipulatability for the user by making available information of the results of searches performed in the past and the current system state. The information search terminal and system includes a query statement input window for inputting a search query statement for a search term, a search history display window for displaying the search query statement and the number of documents as hit in the search, a search result list display window for displaying in juxtaposition a plurality of titles of documents as hit in the form of a list, and a document display window for displaying a document containing the search term and resulting from the search

Patent Number: **5,051,947**
Class: **364** **Date Issued:** **19910924** **Distance:** **0.342793**
SubClass: **900** **Date Filed:** **19851210**
Inventors: **Messenger; Charles H.:Heiss, Jr.; Robert E.**
Assignee: **TRW Inc.**

High-speed single-pass textual search processor for locating exact and inexact matches of a search pattern in a textual stream

A high speed search processor capable of performing a wide variety of search functions, including simple and complex searches, either within an entire text stream or within predefined fixed or sliding windows in the text stream. The processor is made up of multiple interconnected cells, each of which has a pattern register for storing part of a pattern to be searched for, a character register for storing a character of the data stream to be searched, a match register for storing a match value indicative of a match between the search pattern and the text stream, and match logic for modifying an incoming match value in accordance with conditions within the cell. The data stream and the search pattern are oppositely oriented, such that a first character of the search pattern is first encountered by a first character in the data stream, and the pattern is successively compared with an equal number of characters in the data stream as it is moved through the search pattern. The match logic includes means for detecting missing and extra characters in the data stream. The processor can therefore tolerate incorrect, missing or extra characters in the text stream, and can handle multiple levels of nesting and arbitrary boolean expressions within the search pattern. Another novel aspect of the processor is its ability to locate an enumerated subset of search terms or patterns within fixed or sliding windows.

Patent Number: **4,276,597**
Class: **364** **Date Issued:** **19810630** **Distance:** **0.340096**
SubClass: **300** **Date Filed:** **19740117**
Inventors: **Dissly; Donald D.:Blanchard; Ronald J.**
Assignee: **Volt Delta Resources, Inc.**

Method and apparatus for information storage and retrieval

Method and apparatus for identifying particular desired information bearing records having desired predetermined identifiable characteristics from a set of such records in a base data file. A special retrieval file including arrays of binary coded elements is produced and maintained from the information content of the base data file. Each array of the retrieval file corresponds to a particular predetermined identifiable characteristic of language structure potentially present in or associated with the set of records concerned and each element in such an array corresponds to and is representative of the address or location of a particular one of the records in the base data file. The elements are binary coded to represent the presence or absence of the predetermined identifiable characteristics of language structure associated with that particular array in the corresponding record. Furthermore, the set of predetermined identifiable characteristics is itself chosen, in one exemplary embodiment, to represent the alphabetic value and relative sequential location of information characters in associated groups of characters such as words contained in the records. In this manner, the retrieval file itself represents an irreversible information compression of the language structure and/or information contained in the set of information bearing records.
To locate any particular desired record, the retrieval file is first searched by identifying and selecting those arrays representing desired predetermined identifiable characteristics of language structure and comparing the binary values of respectively corresponding elements in the selected arrays thus identifying which records in the base data file have all the desired predetermined identifiable characteristics of language structure. Once the desired records in the base data file have been identified in this manner, they are then selected and displayed, copied, etc., as desired to provide the requisite access or retrieval of information that had previously been stored in the base data file. Particular choices and variations in the selection of the set of predetermined identifiable characteristics of language structure to be represented by the arrays in the retrieval file will change the search and retrieval characteristics, capabilities, flexibility, etc., of the system as may be desired for particular types of record sets and particular types of base data file formats, etc.

Patent Number: **5,175,814**
Class: **395** **Date Issued:** **19921229** **Distance:** **0.337720**
SubClass: **161** **Date Filed:** **19900130**
Inventors: **Anick; Peter G.:Brennan; Jeffrey D.:Flynn; Rex A.:Alvey; Bryan:Hanssen; David R.:Robbins; Jeffrey M.**
Assignee: **Digital Equipment Corporation**

Direct manipulation interface for Boolean information retrieval

A method and apparatus that translates a natural language query into a Boolean expression to be used to search a database. The Boolean expression is displayed on a screen so that the user can alter the Boolean expression using a mouse or similar input device and re-execute the search. The manipulations performed by the user include moving terms of the query, changing the order in which query terms are evaluated, adding terms, deleting terms, and selecting alternate versions of terms.

Patent Number:	**4,939,689**				
Class:	**364**	**Date Issued:**	**19900703**	**Distance:**	**0.337627**
SubClass:	**900**	**Date Filed:**	**19870409**		
Inventors:	**Davis; Mary L.:Rose; David:Barrow; Michael D.**				
Assignee:	**Crowninshield Software, Inc.**				

Outline-driven database editing and retrieval system

A relational database is created and queried through the use of an outliner-style text editor which permits automatic generation of data entry forms for the creation of records. Data entry and editing are simplified and errors are minimized because changes in the outline are automatically reflected in the data entry forms and thus the automatically updated records. Data retrieval is driven through the manipulation of the outline to allow simple and complex queries without utilizing a database programming language. A specialized global field is utilized in which identical field names may be repetitively inserted into several databases. In the data entry mode, a global value can be set and that value is automatically inserted into each database record containing that global field as they are created so that relations are made automatically within the various databases. In the data retrieval mode, the global field can be used to control the display of the outline to truncate the outline to only those categories and fields containing data for a specific global field value, thereby to display only relevant outline portions. A field mapper allows the operator to immediately see the changes in the outline and direct old fields to new names or positions and indicate new fields which are to be inserted into the existing records, all prior to execution of the changed outline in terms of data entry. The query mode features a continually displayed outline in an Outline Window.

Patent Number:	**4,358,824**				
Class:	**364**	**Date Issued:**	**19821109**	**Distance:**	**0.337202**
SubClass:	**200**	**Date Filed:**	**19791228**		
Inventors:	**Glickman; David:Repass; James T.:Rosenbaum; Walter S.:Russell; Janet G.**				
Assignee:	**International Business Machines Corporation**				

Office correspondence storage and retrieval system

A system that intelligently abstracts and archives a document for storage and interprets a free form user retrieval query to recall the document from the storage file. The system includes a method for automatically selecting keywords from the document using a parts of a speech directory. A method is given for weighing the importance or centrality of each keyword with respect to the document of its origin. Using the same logic paths, a free form query that describes the document in the same manner that it would have to be described to a secretary to "find" it in a filing cabinet, the system automatically determines the key matching terms and finds the archived document(s) with the greatest affinity.

Patent Number:	**4,773,038**				
Class:	**364**	**Date Issued:**	**19880920**	**Distance:**	**0.331865**
SubClass:	**900**	**Date Filed:**	**19860224**		
Inventors:	**Hillis; W. Daniel:Lasser; Clifford:Kahle; Brewster:Sims; Karl**				
Assignee:	**Thinking Machines Corporation**				

Method of simulating additional processors in a SIMD parallel processor array

A method is described for simulating additional processors in a SIMD computer by dividing the memory associated with each processor into a plurality of sub-memories and then operating on each sub-memory in succession as if it were associated with a separate processor. Thus, a first instruction or set of instructions is applied to all the processors of the array to cause at least some processors to process data stored at a first location or locations in the first sub-memory. Thereafter, the same first instruction or set of instructions is applied to all the processors of the array to cause at least some processors to process data stored at the same first location in a second sub-memory. And so forth for each of the sub-memories. By operating a SIMD computer in this fashion, it is possible in effect to vary the number of processors in the array so as to provide the number of processors required for a problem.

Patent Number:	**4,451,901**				
Class:	**364**	**Date Issued:**	**19840529**	**Distance:**	**0.331855**
SubClass:	**900**	**Date Filed:**	**19820121**		
Inventors:	**Wolfe; Donald W.:Dye, Jr.; Richard W.**				
Assignee:	**General Electric Company**				

High speed search system

A system for searched digitized data permitting the simultaneous searching of several queries. To minimize excessive matches of queries, the search may be limited to selected portions of each of the documents being searched.

Patent Number: **4,823,306**
Class: **364** **Date Issued:** **19890418** **Distance:** **0.329759**
SubClass: **900** **Date Filed:** **19870814**
Inventors: **Barbic; Federico:Choy; David M.**
Assignee: **International Business Machines Corporation**

Text search system

In a searching for library documents that match the content of a given sequence of query words, a set of equivalent words are defined for each query word along with a corresponding word equivalence value assigned to each equivalent word. Target sequences of words in a library document which match the sequence of query words are located according to a set of matching criteria. The similarity value of each target sequence is evaluated as a function of the corresponding equivalence values of words included therein. Based upon the similarity values of its target sequences, a relevance factor is then obtained for each library document.

Patent Number: **4,383,307**
Class: **364** **Date Issued:** **19830510** **Distance:** **0.328667**
SubClass: **900** **Date Filed:** **19810504**
Inventors: **Gibson, III; Stuart M.**
Assignee: **Software Concepts, Inc.**

Spelling error detector apparatus and methods

A spelling error detector apparatus employs a memory which stores alphabetical words as those existing in the English language in three major memory sections which constitutes a most frequently used word list (MFU), a master word list (MWL) and a personal word list (PWL). Each word is uniquely coded and stored as a 24 bit binary number. The system then retrieves words which are stored or entered into a processor memory and which words are indicative of a document to be printed. Each word emanating from the processor memory is converted into the same code as the stored words and then a search is made to determine whether the processor word compares with a word as stored. If a favorable comparison is had, it is assumed that the spelling of the processor word is correct. If an unfavorable comparison is had, it is assumed that the spelling is incorrect and the misspelled word is stored in a separate memory which can be accessed by the operator in order to make the necessary corrections. Similarly each word which is correctly spelled is also stored in a memory which has the capacity to store a plurality of the last words checked by the system. Based on the system considerations the coding of the words assures a very low collision rate and hence the system is extremely reliable in detecting misspelled words according to the disclosed techniques.

Patent Number: **4,996,666**
Class: **365** **Date Issued:** **19910226** **Distance:** **0.326572**
SubClass: **49** **Date Filed:** **19880812**
Inventors: **Duluk, Jr.; Jerome F.**
Assignee:

Content-addressable memory system capable of fully parallel magnitude comparisons

A content-addressable memory for storing a plurality of words, each word comprising a plurality of data subfields, and each data subfield comprising a plurality of data bits. Query operations simultaneously compare input data to all subfields in all words and selectably test each subfield for either equality, less-than, less-than-or-equal-to, greater-than, greater-than-or-equal-to, inequality, or don't care. A flag memory comprising a plurality of flag bits for each word stores the results of a selectable Boolean operation performed on a set of flag bits and the query results. A mask register causes selected bit positions within words to be treated as not being present. A priority resolver finds the highest priority flag bit in a particular logic state for selecting a word for reading or writing. A content-addressable memory system composed of a plurality of content-addressable memories and an external priority resolver for selecting between content-addressable memories for reading or writing.

Patent Number: **5,109,439**
Class: **382** **Date Issued:** **19920428** **Distance:** **0.325824**
SubClass: **61** **Date Filed:** **19900612**
Inventors: **Froessl; Horst**
Assignee:

Mass document storage and retrieval system

A sequence of documents is delivered to an optical scanner in which each document is scanned to form a digital image representation of the content of the document. In one embodiment, the image representation is converted into code (ASCII) and is automatically examined by data processing apparatus to select search words which meet predetermined criteria and by which the document can subsequently located. In another embodiment, the image is not converted. The search words are stored in a nonvolatile memory in code form and the entire document content is stored in mass storage, either in code or image form. Techniques for selecting the search words are disclosed.

Patent Number: 4,748,439
Class: 340 **Date Issued: 19880531** **Distance: 0.324724**
SubClass: 1462 **Date Filed: 19850813**
Inventors: Robinson; Ian N.:Brunvand; Erik L.:Davis; Alan L.
Assignee: Fairchild Semiconductor Corporation

Memory apparatus and method for retrieving sequences of symbols including variable elements

A memory system and method for the storage and retrieval of sequences of symbols which may be used to represent rules in artificial intelligence systems. The stored data sequence consist of a plurality of symbols, each symbol belonging to one of three classes, constants, variables, or delimiters. Stored data sequences are retrieved by the apparatus of the present invention in response to a query sequence which consists of a plurality of symbols belonging to the same three classes as the symbols comprising the stored data sequences. A stored data sequence is retrieved in response to a given query sequence if the two sequences can be made identical by replacing each variable element appearing in the two sequences by a constant or a combination of constants and delimiters, said combination beginning and ending with a delimiter. Different constants or combinations thereof may be used for each variable element replaced. The apparatus consists of a memory, a means for receiving a query sequence coupled to the apparatus, and a data processing system for comparing the query sequence with each of the stored data sequences and retrieving those data sequences which correspond to the query sequence. The data processing system may be structured so as to contain a plurality of processors working in parallel, each of which operating on a different group of stored data sequence symbols so as to decrease the time needed to find the data sequences corresponding to a given query sequence.

Patent Number: 5,222,234
Class: 395 **Date Issued: 19930622** **Distance: 0.322113**
SubClass: 600 **Date Filed: 19920210**
Inventors: Wang; Diana S.:Kastelic; Francis J.
Assignee: International Business Machines Corp.

Combining search criteria to form a single search and saving search results for additional searches in a document interchange system

A method of saving a search criteria and search results of a document search in a document interchange system having a plurality of shared libraries. The search criteria is stored in a Search Criteria Document appropriately encoded for the document interchange management system. The search results are saved in a Search Result Document. This Search Result Document may be reused or stored in another document such as a folder.

Patent Number: 5,251,131
Class: 364 **Date Issued: 19931005** **Distance: 0.321777**
SubClass: 41908 **Date Filed: 19910731**
Inventors: Masand; Brij M.:Smith; Stephen J.
Assignee: Thinking Machines Corporation

Classification of data records by comparison of records to a training database using probability weights

Classification of natural language data wherein the natural language data has an open-ended range of possible values or the data values do not have a relative order. A training database stores training records, wherein each training record includes predictor data fields. Each predictor data field containes a feature, wherein each feature is a natural language term, and a target data field containing a target value representing a classification of the record. Features may also include conjunctions of natural language terms and each feature may also be a member of a category subset of features. The training database stores, for each feature, a probability weight value representing the probability that a record will have the target value contained in the target data field if a feature contained in a corresponding predictor data field occurs in the record. Features are extracted from a new record and each feature from the new record is used to query the training records to determine the probability weights from the training records having matching features. The probability weights are accumulated for each training record to determine a comparison score representing the probability that the training record matches the new record and provide an output indicating the training records most probability matching the new record.

Patent Number: 5,274,818
Class: 395 **Date Issued: 19931228** **Distance: 0.321762**
SubClass: 700 **Date Filed: 19920203**
Inventors: Vasilevsky; Alexander D.:Sabot; Gary W.:Lasser; Clifford A.:Tennies; Lisa

A.:Weinberg; Tobias M.:Seamonson; Linda J.
Assignee: Thinking Machines Corporation

System and method for compiling a fine-grained array based source program onto a course-grained hardware

The present invention provides a parallel vector machine model for building a compiler that exploits three different levels of parallelism found in a variety of parallel processing machines, and in particular, the Connection Machine Computer CM-2 system. The fundamental idea behind the parallel vector machine model is to have a target machine that has a collection of thousands of vector processors each with its own interface to memory. Thus allowing a fine-grained array-based source program to be mapped onto a course-grained hardware made up of the vector processors. In the parallel vector machine model used by CM Fortran 1.0, the FPUs, their registers, and the memory hiearchy are directly exposed to the compiler. Thus, the CM-2 target machine is not 64K simple bit-serial processors. Rather, the target is a machine containing 2K PEs (processing elements), where each PE is both superpipelined and superscalar. The compiler uses data distribution to spread the problem out among the 2K processors. A new compiler phase is used to separate the code that runs on the two types of processors in the CM-2; the parallel PEs, which execute a new RISC-like instruction set called PEAC, and the scalar front end processor, which executes SPARC or VAX assembler code. The pipelines in PEs are filled by using vector processing techniques along the PEAC instruction set. A scheduler overlaps the execution of a number of RISC operations.

Patent Number: 4,747,072
Class: 364 **Date Issued: 19880524** **Distance: 0.321671**
SubClass: 900 **Date Filed: 19850813**
Inventors: Robinson; Ian N.:Davis; Alan L.
Assignee: Fairchild Camera and Instrument Corporation

Pattern addressable memory

A memory system for storing and retrieving data sequences of symbols in response to a query sequence is disclosed. Each of the sequences is made up of three types of symbols, constants, delimiters, and variables. A data sequence is retrieved in response to a query sequence if the two sequence can be made identical by replacing the variables in each sequence by constants or combinations of constants and delimiters, the combinations beginning and ending with a delimiter. To reduce the time needed to search the memory for all data sequences corresponding to a given query sequence, multiple processing units are employed. In addition to carrying out rule-based searches, the memory system can efficiently retrieve all records containing a specified list of key words.

Patent Number: 4,924,435
Class: 364 **Date Issued: 19900508** **Distance: 0.320558**
SubClass: 900 **Date Filed: 19880502**
Inventors: Brunvand; Eric L.:Davis; Alan L.
Assignee: Fairchild Semiconductor Corporation

Circulating context addressable memory

A memory system for storing and retrieving data sequences of symbols in response to a query sequence. Each of the data sequences and the query sequence is made up of three types of symbols, constants, delimiters, and variables. A data sequence is retrieved in response to a query sequence if the two sequences can be made identical by replacing each variable in each sequence by constants or combinations of constants and delimiters, said combinations beginning and ending with a delimiter. The data sequences are stored in a circulating memory store in which each symbol periodically passes by a number of Tap points at which it is available for reading. Each of the Tap points contains a processor which is capable of comparing the stored data sequences with the query sequence. A unique sub-system is provided for entering new data sequences into the circulating memory store without interrupting the operation of the data retrieval functions. This data entry technique automatically collects fragmented storage areas which were to small to store the new data sequence and combines these into a larger space into which the new data sequence is inserted.

Patent Number: 4,827,403
Class: 364 **Date Issued: 19890502** **Distance: 0.320443**
SubClass: 200 **Date Filed: 19861124**
Inventors: Steele, Jr.; Guy L.:Hillis; W. Daniel:Blelloch; Guy:Drumbeller; Michael:Kahle; Brewster:Lasser; Clifford:Ranade; Abhiram:Salem; James:Sims; Karl
Assignee: Thinking Machines Corporation

Virtual processor techniques in a SIMD multiprocessor array

A virtual processor mechanism and specific techniques and instructions for utilizing such virtual processor mechanism within an SIMD computer having numerous processors, and each physical processor having dedicated memory associated therewith. Each physical processor is used to simulate multiple "virtual" processors, with each physical processor simulating the same number of virtual processors. The memory of each physical processor is divided into n regions of equal size, each such region being allocated to one virtual processor, where n is the number of virtual processors simulated by each physical processor. Whenever an instruction is processed, each physical processor is time-sliced among the virtual memory regions, performing the operation first as one virtual processor, then another, until the operation has been performed for all virtual processors. Physical processors are switched among the virtual processors in a completely regular, predictable, deterministic fashion. The virtual processor mechanism switches among virtual processors within instructions, so that at the completion of each instruction, it has been executed on behalf of all virtual processors. A number of instructions are shown for execution using these virtual processor techniques.

Patent Number: 5,050,071
Class: 364 **Date Issued: 19910917** **Distance: 0.318395**
SubClass: 200 **Date Filed: 19881104**
Inventors: Harris; Edward S.:Kleinberger; Paul:Blanks; Natan:Kraus; Richard B.:Wolfe; Thomas R.
Assignee:

Text retrieval method for texts created by external application programs

A computer program for the storage and retrieval to information texts is provided the ability to access and retrieve external text files created by other application programs. The program stores information as texts and creates a summary text associated with each external text file. Keywords associated with the external text can then be indexed to the summary text so that a keyword search of the information texts can use the summary text. If a search leads to the summary text, the user can use a hot link to immediately launch the external program to access the external text file.

Patent Number: 3,909,796
Class: 340 **Date Issued: 19750930** **Distance: 0.317901**
SubClass: 1725 **Date Filed: 19730921**
Inventors: Kitamura; Tetsuo
Assignee: Ricoh Co., Ltd.

Information retrieval system serially comparing search question key words in recirculating registers with data items

Key words defining a search question are entered into circulating registers which circulate the key words in synchronism with data units read out serially from a storage device, each data unit consisting of a key word and codes for locating documents or bibliographies corresponding to said key word. The key words of the data units read out from the storage device are compared with the search question key words in the registers. The data unit whose key word coincide with search question key words are transmitted to a secondary information retrieval device such as a microfilm reader-printer so that the corresponding documents can be retrieved and displayed.

Patent Number: 5,170,370
Class: 364 **Date Issued: 19921208** **Distance: 0.316569**
SubClass: 736 **Date Filed: 19901121**
Inventors: Lee; William:Geissler; Gary J.:Johnson; Steven J.:Schiffleger; Alan J.
Assignee: Cray Research, Inc.

Vector bit-matrix multiply functional unit

A method and apparatus provides bit manipulation of data in vector registers of a vector register computer system. Matrix multiplication is accomplished at a bit level of data stored as two matrices in a vector computer system to produce a matrix result. The matrices may be at least as large as 64 bits by 64 bits and multiplied by another 64 by 64 matrix by means of a vector matrix multiplication functional unit operating on the matrices within a vector processor. The resulting data is also stored at a 64 bit by 64 bit matrix residing in a resultant vector register.

Patent Number: 5,021,945
Class: 364 **Date Issued: 19910604** **Distance: 0.316227**
SubClass: 200 **Date Filed: 19890626**
Inventors: Morrison; Gordon E.:Brooks; Christopher B.:Gluck; Frederick G.
Assignee: MCC Development, Ltd.

Parallel processor system for processing natural concurrencies and method therefor

A computer processing system containing a plurality of identical processor elements each of which does not retain execution state information from prior operations. The plurality of identical processor elements operate on a statically compiled program which, based upon detected natural concurrencies in the basic blocks of the programs, provide logical processor numbers and an instruction firing time to each instruction in each basic block. Each processor element is capable of executing instructions on a per instruction basis such that dependent instructions can execute on the same or different processor elements. A given processor element is capable of executing an instruction from one context followed by an instruction from another context through use of shared storage resources.

Patent Number: 4,933,895
Class: 364 **Date Issued: 19900612** **Distance: 0.315937**
SubClass: 748 **Date Filed: 19870710**
Inventors: Grinberg; Jan:Nash; James G.:Little; Michael J.
Assignee: Hughes Aircraft Company

Cellular array having data dependent processing capabilities

A cellular array processor (10) for efficiently performing data dependent processing such as floating point arithmetic functions. One module (84) in the array processor (12) generates a signal applied to bus line (24) when all of the bits in a register (86) are zero. The signal on bus line (24) effects the shifting operation of a shift register (36) in a memory module (34) located on a different functional plane. Thus, the processing functions carried out in each elemental processor (26) can be made to depend on the value of data stored therein instead of being dictated solely by a simultaneous executed instruction from the control processor (14) as is the normal case.

Patent Number: 5,268,856
Class: 364 **Date Issued: 19931207** **Distance: 0.314712**
SubClass: 748 **Date Filed: 19920925**
Inventors: Wilson; Stephen S.
Assignee: Applied Intelligent Systems, Inc.

Bit serial floating point parallel processing system and method

A system and method for floating point computations involving matrices or vectors includes a plurality of identical processing units connected to a linear chain with direct data communication links between adjacent processing units. Each such processor is also connected to its own private memory. A sequence of instructions is sent by a controller to all floating point processing units and their associated memories whereby all processing units in the chain receive the same instruction and all memories receive the same address at any given cycle in the instruction sequence. Each processing unit internally handles floating point operations such as normalization, sign changes, and multiplication in a bit serial manner.

Patent Number: 5,050,075
Class: 364 **Date Issued: 19910917** **Distance: 0.313777**
SubClass: 200 **Date Filed: 19881004**
Inventors: Herman; Gary E.:Lee; Kuo-Chu:Matoba; Takako
Assignee: Bell Communications Research, Inc.

High performance VLSI data filter

A single chip high speed VLSI data filter is disclosed. The data filter performs relational and simple numeric operations on a high speed input data stream using a unique instruction set containing no branching instructions.

Patent Number: 3,970,993
Class: 340 **Date Issued: 19760720** **Distance: 0.313532**
SubClass: 1725 **Date Filed: 19740102**
Inventors: Finnila; Charles A.
Assignee: Hughes Aircraft Company

Cooperative-word linear array parallel processor

A "cooperative-word" linear array parallel processor comprises many logically identical memory words or microprocessors ordered in a linear array by a Chaining channel. Inasmuch as the Chaining channel can contain different information (either data to be processed or control information) at each word position, it permits highly parallel word-cooperative operations such as pair-wise arithmetic. The processor also has several global communication channels in which data may be transferred between an external buffer and a specified subset of processor words. Inasmuch as individual words may be addressed by their content rather than by their physical locations, relatively simple switching logic within each word provides effective self-repair. A plurality of flag flip-flops in each individual cell interact with gobal control lines to activate processing within a particular cell and to indicate the results of operations performed by that cell. Flag data can be passed from one word to another by means of the chaining channel, or they can be manipulated within a word by means of the aforesaid global control lines.

Infringement Search Report

Patent Number: **4,941,125**
Class: **364** **Date Issued:** **19900710** **Distance:** **0.312296**
SubClass: **900** **Date Filed:** **19840801**
Inventors: **Boyne; Walter J.**
Assignee: **Smithsonian Institution**

Information storage and retrieval system

A digital camera is used to scan documents and generate a corresponding digital output signal. A data processor receives the digital output signal and generates corresponding index information. The video and index information are then stored on one or more optical disks for search and retrieval.

Patent Number: **5,159,180**
Class: **235** **Date Issued:** **19921027** **Distance:** **0.312055**
SubClass: **375** **Date Filed:** **19900919**
Inventors: **Feiler; William S.**
Assignee:

Litigation support system and method

A litigation support system and method in which information regarding documents and other items of evidence are stored in record fields of an electronic database using an optical scanning mechanism with the ability to scan bar-codes or other indicia. By using a bar-code to generate a single relation among databases as well as using bar-code authority lists for entries into the disclosed litigation support computer system, significant improvement in both the speed of coding documents or other items of evidence as well as the accuracy of such coding is greatly enhanced.

Patent Number: **4,760,523**
Class: **364** **Date Issued:** **19880726** **Distance:** **0.310397**
SubClass: **200** **Date Filed:** **19861224**
Inventors: **Yu; Kwang-I:Hsu; Shi-Ping:Hasiuk; Lee Z.:Otsubo; Peggy M.**
Assignee: **TRW Inc.**

Fast search processor

A special-purpose search processor, and a related method, for performing a variety of logically complex searches of a serial data stream in a highly concurrent fashion. The processor comprises a sequence of serially connected cells of identical construction, and the data stream is passed through the sequence of cells, each cell performing a logical operation based only on the data provided to it from the previous cell in the sequence. Each cell has a character register for data storage and a pattern register for storage of part of a search pattern. The contents of the two registers are compared in each cell, at each cycle of a clock used to propagate the data through the processor. Match indicators or match tolerance values are propagated through the processor on a match line, and match results emerge in synchronism with the data stream. Multiple match lines are employed in one preferred embodiment, to temporarily save, retrieve and exchange match tolerance values, in order to effect logically complex searches in a highly concurrent manner. Types of searches that may be performed include logical OR and AND searches, common-prefix OR searches, and searches involving variable-length and fixed-length don't-care strings, variable-length care strings, and negate strings.

Patent Number: **4,876,643**
Class: **364** **Date Issued:** **19891024** **Distance:** **0.309861**
SubClass: **200** **Date Filed:** **19870624**
Inventors: **McNeill; Kevin M.:Ozeki; Takeshi**
Assignee: **Kabushiki Kaisha Toshiba**

Parallel searching system having a master processor for controlling plural slave processors for independently processing respective search requests

A parallel processing search system for searching and updating a database at the request of a host system, including a master processor connected to a host system bus for transfer of information between said master processor and the host system bus; a data bus connected to the master processor; plural slave processors connected to the data bus for independently processing search respective requests under the control of the master processor; a disk drive interface adapted to be connected to a disk which stores a database; and a buffer memory connected to the data bus and the disk drive for storing the database retrieved from the disk and for sequentially placing data from the database on the data bus for match comparison by the slave processors so that a search of the database can be made by the slave processors under the control of the master processor. The buffer memory is also capable of storing updated data obtained from the host system via the master processor so that an updated database can be transferred to the disk memory via the disk drive interface.

Patent Number: 4,255,796
Class: 364 Date Issued: 19810310 Distance: 0.309625
SubClass: 900 Date Filed: 19780214
Inventors: Gabbe; John D.:Judice; Charles N.:London; Thomas B.
Assignee: Bell Telephone Laboratories, Incorporated

Associative information retrieval continuously guided by search status feedback

An associative information retrieval system accepts information from a user and generates a query mask utilizing nested superimposed code words to search through and to find partial matches with the content of an auxiliary store. The auxiliary store contains similarly generated code words each produced from attribute values of records on a central store. The user information is put through the system on a character-by-character basis and the user is fed back information on the number of possible matches. The feedback informs the user on the incremental progress of the search produced in response to each newly entered character and also as part of a sequence that it may form with previously entered characters. The feedback information helps the user direct the search which the person does by supplying additional characters. When the number of possible matches is reduced to a manageable list, the index codes associated with the partially matching stored code words in the auxiliary store are used to locate complete records from a central store for display to the user to complete the retrieval process. In this latter process, false drops are eliminated by matching characters used to form the query mask directly with those of the records which were located via the nested superimposed code words.

Patent Number: 4,839,853
Class: 364 Date Issued: 19890613 Distance: 0.308511
SubClass: 900 Date Filed: 19880915
Inventors: Deerwester; Scott C.:Dumais; Susan T.:Furnas; George W.:Harshman; Richard A.:Landauer; Thomas K.:Lochbaum; Karen E.:Streeter; Lynn A.
Assignee: Bell Communications Research, Inc.

Computer information retrieval using latent semantic structure

A methodology for retrieving textual data objects is disclosed. The information is treated in the statistical domain by presuming that there is an underlying, latent semantic structure in the usage of words in the data objects. Estimates to this latent structure are utilized to represent and retrieve objects. A user query is recouched in the new statistical domain and then processed in the computer system to extract the underlying meaning to respond to the query.

Patent Number: 5,008,815
Class: 364 Date Issued: 19910416 Distance: 0.307183
SubClass: 200 Date Filed: 19880627
Inventors: Hillis; W. Daniel
Assignee: Thinking Machines Corporation

Parallel processor

A parallel processor array is disclosed comprising an array of processor/memories and means for interconnecting these processor/memories in an n-dimensional pattern having at least 2n nodes through which data may be routed from any processor/memory in the array to any other processor/memory. Each processor/memory comprises a read/write memory and a processor for producing an output depending at least in part on data read from the read/write memory and on instruction information. The interconnecting means comprises means for generating an address message packet that is routed from one processor/memory to another in accordance with address information in the message packet and a synchronized routing circuit at each node in the n-dimensional pattern for routing message packets in accordance with the address information in the packets. Preferably the address information in the message packet is relative to the node in which the message packet is being sent and each digit of the address represents the relative displacement of the message packet in one dimension from the node to which the message packet is being sent. Advantageously, the n-dimensional pattern is a Boolean cube of 15 dimensions. With presently available technology, more than one million such processor/memories can be operated in parallel while interconnected by these interconnecting means.

Infringement Search Report

Patent Number: 5,210,870
Class: 395 **Date Issued: 19930511** **Distance: 0.305814**
SubClass: 600 **Date Filed: 19900327**
Inventors: Baum; Richard I.:Brent; Glen A.:Gibson; Donald H.:Lindquist; David B.
Assignee: International Business Machines

Database sort and merge apparatus with multiple memory arrays having alternating access

A processor functioning as a coprocessor attached to a central processing complex provides efficient execution of the functions required for database processing: sorting, merging, joining, searching and manipulating fields in a host memory system. The specialized functional units: a memory interface and field extractor/assembler, a Predicate Evaluator, a combined sort/merge/join unit, a hasher, and a microcoded control processor, are all centered around a partitioned Working Store. Each functional unit is pipelined and optimized according to the function it performs, and executes its portion of the query efficiently. All functional units execute simultaneously under the control processor to achieve the desired results. Many different database functions can be performed by chaining simple operations together. The processor can effectively replace the CPU bound portions of complex database operations with functions that run at the maximum memory access rate improving performance on complex queries.

Patent Number: 5,265,242
Class: 395 **Date Issued: 19931123** **Distance: 0.305195**
SubClass: 600 **Date Filed: 19871230**
Inventors: Fujisawa; Hiromichi:Hatakeyama; Atsushi:Nakano; Yasuaki:Higashino; Junichi:Hananoi; Toshihiro
Assignee:

Document retrieval system for displaying document image data with inputted bibliographic items and character string selected from multiple character candidates

A document storage and retrieval system for storing a document body in the form of image, means for storing text information in the form of a character code string for retrieval, apparatus for executing a retrieval with reference to the text information, and apparatus for displaying a document image relating thereto on a retrieval terminal according to the retrieval result. Such a form of the system is available for retrieving the full contents of a document and also for displaying the document body printed in a format easy to read straight in the form of image. Users are capable of retrieving documents with arbitrary words and also capable of reading even such a document as is complicated to include mathematical expressions and charts through a terminal in the form of image, the same as on paper. A system is provided wherein the text information for retrieval is extracted automatically from the document image through character recognition. Since a precision of the character recognition has not been satisfactory hitherto, a visual retrieval and correction have been carried out without fail by operators. However, there is no necessity for the operators to attend therefor.

Patent Number: 4,241,402
Class: 364 **Date Issued: 19801223** **Distance: 0.305104**
SubClass: 200 **Date Filed: 19781012**
Inventors: Mayper, Jr.; Victor:Nagy; Alex L.:Bird; Richard M.:Tu; Ju Ching:Michels; Lowell S.
Assignee: Operating Systems, Inc.

Finite state automaton with multiple state types

The subject of this disclosure is a Finite State Automaton (FSA) used as part of a term detector employed in a digital pattern search system (searcher). In particular the invention includes various advances in the art of FSA design which make the FSA practical for pattern recognition.
Specifically, these advances minimize the amount of memory which is required in each FSA in performing pattern recognition, and allow a speed capability such that the searching can be performed at the rate at which a mass storage medium can supply data. The large amount of memory required and the low speed of processing in the prior state of the art made the use of an FSA impractical for most real applications.
The new advances include the following:
An adaptation of the indexing means described in reference 4, which allows simple selection of the correct success transition state from a number of possible success states;
The partitioning of searchable digital patterns into parts (called nibbles) to reduce the amount of memory used within an FSA;
The use of various types of states to allow the detection of specific input patterns in the presence of don't-care patterns;
The use of multiple FSA's to reduce the amount of memory needed in these FSA's when handling don't-care patterns;
The unique design of an FSA to search for multiple sequential input patterns;
The unique features of the FSA design to allow recognition of numerical ranges of values from among numerical

data.